

5

**U.S. PATENT APPLICATION ENTITLED:**

10      **SYSTEM, METHOD AND ARTICLE OF**  
**MANUFACTURE FOR FACILITATING**  
**USER INTERFACE ROAMING IN AN**  
**INTERFACE ROAMING NETWORK**  
**FRAMEWORK**

15

**INVENTORS:**

Christian V. Olgaard  
Spiros N. Bouas  
Rick W. Umstattd  
Benny Madsen

20

25

30      **BAKER & MCKENZIE**  
Two Embarcadero Center  
Twenty-fourth Floor  
San Francisco, California 94111-3909  
Telephone (415) 576-3000

# **SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR FACILITATING USER INTERFACE ROAMING IN AN INTERFACE ROAMING NETWORK FRAMEWORK**

5

## **INVENTORS:**

Christian V. Olgaard  
Spiros N. Bouas  
Rick W. Umstattd  
Benny Madsen

10

## **FIELD OF THE INVENTION**

The present invention relates generally to hybrid networks and, more particularly, to a  
network framework for facilitating interface roaming.

15

## **BACKGROUND OF THE INVENTION**

As wireless communication is moving from being voice centric to data centric, the  
need to move towards devices that are more integrated are being pursued. People  
are creating smart phones, which integrate PDA functionality and a larger display  
with a cell phone, or people are trying to integrate cell phone (voice & data)  
functionality into a pocket sized PDA. The goal of most of these efforts is to offer  
the user a wireless Internet experience. However, this integration effort is difficult,  
as the user on the one expects a small form factor cell phone, but at the same time  
expect a full graphical internet experience when surfing the net, and a large display  
is required to offer an acceptable experience. Therefore, existing integration efforts  
are making a trade off between size and graphical experience (display size).

20

25

### SUMMARY OF THE INVENTION

A system, method and article of manufacture are provided for facilitating user interface roaming in an interface roaming network framework. Received from a wireless link is a list of usable interface clients in proximity to the wireless link. Each usable interface client has functional capabilities associated with it. One of the interface clients is selected from the list and the wireless link is notified of the selected interface client. A connection is then initiated with the selected interface client and an application is executed based on the capabilities of the selected interface client in order to generate content. The generated content is then transmitted to the interface client so that the content may be displayed using the interface client.

In an embodiment of the present invention, a notification may be received from the wireless link upon activation indicating that the wireless link has been activated. Upon activation, the wireless link may determine the usable interface clients in its proximity. In another embodiment of the present invention, information may be received that was input by the user in response to the generated content.

Subsequent content may then be generated based on the received information and then transmitted to the interface client for display. In one aspect of such an embodiment, the information from the user may be input into the wireless link which then transmits the information to the server.

In an aspect of the present invention, the connection with the selected interface client may be initiated via the wireless link. In another aspect of the present invention, the generated content may be transmitted to the interface client via the wireless link. In a further aspect of the present invention, each interface client includes a visual display for displaying the content. In yet another aspect of the present invention, the wireless link may receive the content if no usable interface clients are proximate to the wireless link (i.e., the list of usable interface clients includes zero usable

interface clients in proximity to the wireless link). In an additional aspect of the present invention, the wireless link and the interface client are may be capable of communicating utilizing Internet protocols such as TCP/IP or IPX protocols. In a further aspect of the present invention, the wireless link may have telephony capabilities so that it may be used as a mobile telephone.

In one embodiment of the present invention, a notification may be received that another or second interface client better suited for displaying the content is proximate to the wireless link. The user may then be queried as to whether the user would like to switch to the second interface client or remain with the currently used interface client. In one aspect of the such an embodiment, the query may be displayed on the current interface client. If a response to the query is received from the user indicating that the user wants to switch to the second interface client, subsequently generated content is then generated based on the capabilities of the second interface client and then transmitted to the second interface client for display thereon.

In yet a further embodiment of the present invention, a user may input information into the wireless link whereupon the information is only transmitted to the server by the wireless link when a connection between the server and the wireless link is available. In another embodiment of the present invention, execution of an application based on capabilities of the selected interface client may further require uploading from a data store information relating to configuring the application based on the capabilities of the selected interface client.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The foregoing and other features, aspects and advantages are better understood from the following detailed description, appended claims, and accompanying drawings  
5 where:

Figure 1 is a schematic diagram of an interface roaming network framework in accordance with an embodiment of the present invention;

10 Figure 2 is a flowchart of a process for facilitating user interface roaming in an interface roaming network framework in accordance with an embodiment of the present invention;

15 Figure 3 is a flowchart of a process for utilizing a wireless link in an interface roaming network in accordance with an embodiment of the present invention;

Figure 4 is a flowchart for a process for utilizing an interface client in an interface roaming network in accordance with an embodiment of the present invention;

20 Figure 5 is a schematic diagram illustrating a normal mode of operation in accordance with an embodiment of the present invention;

25 Figure 6 is a schematic diagram illustrating a mode of operation where no external interface client is available in accordance with an embodiment of the present invention;

Figure 7 is a schematic diagram illustrating a mode of operation where there is no infrastructure connection from the wireless link to the infrastructure server but where there is access to an interface client proximal to the wireless link in accordance with  
30 an embodiment of the present invention;

Figure 8 is a schematic diagram of a mode of operation wherein there is no connection to infrastructure from the wireless link and no external interface client access in accordance with an embodiment of the present invention;

- 5 Figure 9 is a schematic diagram of some possible exemplary modes for interface roaming when the interface client has its own connection to the server in accordance with an embodiment of the present invention;

- 10 Figure 10 is a schematic diagram of an illustrative system with a plurality of components in accordance with an embodiment of the present invention;

Figure 11 is a schematic diagram of a representative hardware environment in accordance with an embodiment of the present invention.

Figure 11 is a schematic diagram of a representative hardware environment in accordance with an embodiment of the present invention.

### **DETAILED DESCRIPTION**

User interface roaming may generally be defined as the ability to move from one interface (input and output device) and move to another interface while either: (1) maintaining the status of the application(s) and being able to continue from where the previous device was left – even if the interface capabilities change, or (2) changing the status of the application(s) and being able to use a new application based on the capabilities of the new interface client.

10 To solve the problem of having a small cell phone be able to create a sufficient graphical interface to enable a web surfing experience, a system is disclosed that enables interface roaming. Users utilize interface devices (“interface clients”) external to the actual communication engine. This enables use of a small communication device (such as a mobile phone or a wireless personal digital assistant (“PDA”)) while maintaining the ability to present a graphical interface on a display separate from the device. The interface client may be any display which is close or proximate to the user and the user’s communication device. The system will select the best-suited display available to the user.

20 At least two embodiments exist: (1) an embodiment where a connection between an interface client and an infrastructure server is made via a wireless link; and (2) an embodiment where a pre-existing connection exists between an interface client and an infrastructure server.

25 In an embodiment where the connection is made between an interface client and an application via a wireless link, a user carries a wireless link (such as, for example, a mobile phone-like device) and this wireless link creates a connection to the interface client near the user and creates a connection from the wireless link to the application or infrastructure server. In such an embodiment, the wireless link may be considered to act as a gateway with added functionality. The handling of the display of an application on the interface is primarily controlled from and by the infrastructure

server. In a preferred embodiment, the wireless link may include personal identification information associated with the user and provide ways to encrypt the data to different extends, for example, from simple encryption to offering a wireless VPN connection.

5

In an embodiment where a connection between an interface client and an infrastructure server is made using an existing link, the interface client is already connected (wired or wireless) directly to the infrastructure server without communication through the wireless link. In such an embodiment, the user does not need a wireless link since the interface client can already make a connection to the infrastructure server. However, in such an embodiment, a user-identification associated with the user may be required to identify the user to the system. The user-identification may be provided as part of the wireless link's own electronic identification, or any other usable identification method. As a further feature in this embodiment, the interface client can communicate to the infrastructure server not only through its own link, but it can also go through the user's wireless link if available (for example if highly sensitive data must be transferred).

10

15

20

In general applications are primarily executed by the infrastructure server. This makes an interface client serve as an intelligent thin client, that performs the interface translation functions of the application run by the infrastructure server. It is the infrastructure server that modifies the interface appearance to match the capabilities of the selected interface client. What makes this possible is a protocol that allows the interface client to communicate to the infrastructure server via the most suitable connection available and visa versa.

25

Figure 1 is a schematic diagram of an interface roaming network framework 100 in accordance with an embodiment of the present invention. In this framework 100, a wireless link 102 is provided which may be connected to an infrastructure server 104 via a network 106. The wireless link 102 may (or may not) include a visual display and an input device may also have (optionally) telephony capability. The network

30



**106** may be any sort of network, such as a LAN, a WAN, or even the Internet, and may include a wireless network portion **108** which is in communication with the wireless link.

5 The framework **100** also includes a plurality of variously located interface clients **110a, 110b, 110c, 110d, 110e, 110f, 110g**. An interface client includes a processor and preferably has some sort of visual display (e.g. display **112**) such as, for example, a monitor or a television. An interface client may also include some sort of input device (e.g., keyboard **114**) for inputting information and commands. Some of  
10 the interface clients (e.g., interface clients **110d, 110g**) in the framework **100** may also be directly connected to the network **106** to permit communication via the network between the interface client and the infrastructure server **104**.

Each interface client preferably also includes a transceiver **116** for communicating  
15 with the wireless link (and the infrastructure server via the wireless link) by, for example, infrared or RF signals. The wireless link may also have a proximal range **118** within which interface clients (e.g., interface clients **110a, 110b, 110c, 110d**) are considered within the proximity of the wireless link and outside of which interface clients (e.g., interface clients **110e, 110f, 110g**) are considered outside the  
20 proximity of the wireless link. It should be noted that communication between the wireless link and interface clients need not be limited to interface clients inside the proximal range – the range of communication between the wireless link and an interface client may be greater than the proximal range. The proximal range is simply a range inside of which interface clients are considered proximate to the  
25 wireless link (and thereby proximate to a user of the wireless link).

With continuing reference to Figure 1, the interface server **104** may also be connected to a data store/database **120** in which application data and configuration data relating to the various types of interface clients may be stored.

Figure 2 is a flowchart of a process 200 for facilitating user interface roaming in an interface roaming network framework in accordance with an embodiment of the present invention. In operation 202, a list of usable interface clients in proximity to a wireless link is received from the wireless link. Each usable interface client has functional capabilities associated with it. In operation 204, one of the interface clients is selected from the list and the wireless link is notified of the selected interface client in operation 206. A connection is then initiated with the selected interface client in operation 208 and an application is executed based on the capabilities of the selected interface client in order to generate content in operation 210. In operation 212, the generated content is then transmitted to the interface client so that the content may be displayed using the interface client.

In an embodiment of the present invention, a notification may be received from the wireless link upon activation indicating that the wireless link has been activated. Upon activation, the wireless link may determine the usable interface clients in its proximity. In another embodiment of the present invention, information may be received that was input by the user in response to the generated content. Subsequent content may then be generated based on the received information and then transmitted to the interface client for display. In one aspect of such an embodiment, the information from the user may be input into the wireless link which then transmits the information to the server.

In an aspect of the present invention, the connection with the selected interface client may be initiated via the wireless link. In another aspect of the present invention, the generated content may be transmitted to the interface client via the wireless link. In a further aspect of the present invention, each interface client includes a visual display for displaying the content. In yet another aspect of the present invention, the wireless link may receive the content if no usable interface clients are proximate to the wireless link (i.e., the list of usable interface clients includes zero usable interface clients in proximity to the wireless link). In an additional aspect of the present invention, the wireless link and the interface client are may be capable of

communicating utilizing Internet protocols such as TCP/IP or IPX protocols. In a further aspect of the present invention, the wireless link may have telephony capabilities so that it may be used as a mobile telephone.

5 In one embodiment of the present invention, a notification may be received that another or second interface client better suited for displaying the content is proximate to the wireless link. The user may then be queried as to whether the user would like to switch to the second interface client or remain with the currently used interface client. In one aspect of the such an embodiment, the query may be  
10 displayed on the current interface client. If a response to the query is received from the user indicating that the user wants to switch to the second interface client, subsequently generated content is then generated based on the capabilities of the second interface client and then transmitted to the second interface client for display thereon.

15 In yet a further embodiment of the present invention, a user may input information into the wireless link whereupon the information is only transmitted to the server by the wireless link when a connection between the server and the wireless link is available. In another embodiment of the present invention, execution of an  
20 application based on capabilities of the selected interface client may further require uploading from a data store information relating to configuring the application based on the capabilities of the selected interface client.

Figure 3 is a flowchart of a process 300 for utilizing a wireless link in an interface  
25 roaming network in accordance with an embodiment of the present invention. A wireless link is utilized in operation 302 to scan a vicinity of the wireless link to detect one or more interface clients in the vicinity. In operation 304, the wireless link then transmits to an infrastructure server information relating to the interface clients detected in the vicinity. Based on the transmitted information, the  
30 infrastructure server then selects one of the interface clients. Subsequently, the wireless link receives a notification from the infrastructure server of the selected

interface client in operation 306 and, in operation 308, a connection between the infrastructure server and the selected interface client is initiated for communication therebetween.

5 In an embodiment of the present invention, the wireless link may, upon activation, transmit an activation notification to the infrastructure server prior to scanning the vicinity for interface clients to indicate to the infrastructure server that the wireless link is activated. In another embodiment of the present invention, the wireless link may periodically scan the vicinity utilizing the wireless link to detect additional  
10 interface clients, and, if an additional interface client is detected in the vicinity of the wireless link, notify the infrastructure server of the detection of the additional interface client(s). Subsequently, the wireless link may receive a query from the infrastructure server as to whether the user of the wireless link wants to connect to the additional interface client. The wireless link may include a display for displaying  
15 the query to the user and an input device for permitting the user to input an answer to the query for subsequent transmission from the wireless link to the infrastructure server.

In an aspect of the present invention, wherein the infrastructure server and the  
20 selected interface client are connected for communication via the wireless link. In an additional aspect of the present invention, the scanning of the vicinity to detect one or more interface clients may include receiving a signal from each interface client in the vicinity of the wireless link. In another embodiment of such an aspect, the wireless link may first transmit an initial signal in the vicinity for receipt by  
25 interface clients in the vicinity whereby the signals received by the wireless link from the interface clients in the vicinity of the wireless link are transmitted by the interface clients in response to the initial signal from the wireless link.

In a further aspect of the present invention, the information relating to the interface  
30 clients detected in the vicinity may include information relating the capabilities of the detected interface clients. The information may also include information relating

to the locations of the detected information. In one embodiment, the information may be transmitted to the infrastructure server from the wireless link is provided to the wireless link by the interface clients. In yet another embodiment of the present invention, formatting information may be received by the wireless link from the infrastructure server. The wireless link may then subsequently receive content from the infrastructure server. The wireless link may format the content based on the received formatting information and then transmit the formatted information to the interface client so that the formation information may be displayed by the interface client. As an option, the formatted content may be encrypted by the wireless link prior to transmission to the interface client so that only the interface client can access the formatted content.

Figure 4 is a flowchart for a process **400** for utilizing an interface client in an interface roaming network in accordance with an embodiment of the present invention. In general, an interface client in the interface roaming network submits information about the interface client to a wireless link in vicinity to the interface client in operation **402**. The interface client may then subsequently receive content in operation **404** and display the content on a display in operation **406**.

In an embodiment of the present invention, an interface client may receive a signal from the wireless link when the wireless link is in the vicinity of the interface client prior to submitting the information about the interface client to the wireless link. In one such embodiment, the signal from the wireless link may be transmitted from the wireless link in response to a prior signal transmitted from the interface client. Also, the signal from the wireless link may include information identifying a user of the wireless link.

In an aspect of the present invention, the information about the interface client may include: information about the capabilities of the interface client, information about display capabilities of the interface client, information about an input device of the interface client, and/or information about the location of the interface client. In

another embodiment of the present invention, the content may be received from an infrastructure server. In one such embodiment, the content may be received from the infrastructure server via the wireless link. In another such embodiment the interface client may be connected to the infrastructure server via its own connection and not through the wireless link. In another aspect of the present invention, the content may be formatted based on the submitted information about the interface client.

#### Connection Via A User's Wireless Link

As mentioned above, a personal device 102 may be carried by a user to perform a link between the interface client close to the user and the infrastructure. By separating the display and input interface from this device (possibly leaving a small simple display on the device), it will enable a desired small form-factor of the user's device. One may think of this as an extended cell phone, but it may likely be more like a gateway or router that communicate with the wireless infrastructure that also includes a cell phone functionality. The cell phone operation may just be one of many functions that the device serve, and the actual cell phone functionality may be implemented as a wireless head set communicating with the communications device. In the following portion of the specification, the wireless link device 102 may be denoted as a "LitePoint" or simply as a "wireless link".

A functionality of this device is be its ability to roam with different interface clients that it connects to using a short-range wireless connection. This functionality may be implemented with hardware creating the wireless link (the wireless link) and by a protocol that can service roaming and can manage a multiple of interface clients.

When a wireless link 102 is activated, it will connect to the infrastructure server 104 to inform of its activation and also scan for usable interface clients in its proximity. As mentioned earlier, the wireless link may include a rudimentary interface. This interface may only offer minimum functionality and may be similar to a graphic enabled display of a cell phone. In use, a wireless link periodically searches to find available interface clients in its vicinity. It then generate a list of the possible

interface clients and reports this list to the infrastructure server **104**. This list may include various information about the interface client, its display capabilities, its input devices, and the device's geographical location. The infrastructure server then recommends/selects one of the interface clients, and the wireless link contacts the interface client and initiates a connection. From this point on, the user of the wireless link device is able to interact with the interface client. Since the infrastructure server knows the capabilities of the interface client, the infrastructure server can format the user experience to make best use of the interface client's capabilities.

In one embodiment, the infrastructure server may run the application including the display and only transmit the data to the interface client. In another embodiment, the infrastructure server may upload a small program (script, client, etc.) either to the wireless link or to the actual interface client. The infrastructure server then communicates with this program using a low bandwidth protocol, and the program (e.g., interface driver program) formats the infrastructure server generated contents for the interface client.

It is likely that the user experience may change significantly as the interface changes. For example, if one is composing an email on a full screen graphical environment like a normal computer, the user interface is expected to be similar to the well-known computer user interface. If a smaller screen is used and no keyboard is available, it may change to a pen input based interface (the pen input portion may even reside on the wireless link), and if the device is the minimal interface of the wireless link, the user interface may change to become voice operated. Since the interface client reports to the infrastructure server its capabilities, the infrastructure server can upload different interface driver programs to the device to adapt to the user interface if needed. As an option, the most used interface driver programs may reside in the wireless link **102** or an interface client. Since the infrastructure server has access to the wireless link and the interface client, it has the ability to update the interface driver program if needed.

In a preferred embodiment, the applications are running on the infrastructure server, and the wireless link operates more like a thin client device, that still has the ability to execute programs and store data on its own. The wireless link may communicate with the infrastructure server using any high-speed wireless connection, but a protocol for this communication may be designed to enable the described operation, and at the same time geared towards low bandwidth and acceptable latency performance. Similarly, the connection between the wireless link and the external interface client can be any wireless protocol that can provide the sufficient throughput and latency requirement. A standard like Bluetooth or a derivative of this may suffice for the communication between the wireless link and the interface client.

An important aspect of the system may be its ability to roam between interface clients, and maintain the exact point the session was terminated. For example, the user may be receiving stock marked news via voice (i.e., text2voice on the infrastructure server) using the build-in user interface on the wireless link. When a better external interface client is in reach, the infrastructure server may ask the user if he/she wants to switch to the other interface client. As this interface client may allow the user to select specific news-stories using a graphical interface, it is likely that the user may want to switch. The switch may be initiated such that the user may be viewing the story that was presented by the text-voice on the infrastructure server.

The cursor may even be exactly where the reading was terminated at the time of the switch. To make this possible the wireless link may need maintain some information about the application presented to the interface client. If there are no significant changes in the interface client, the system may automatically be able to continue, but if a major change in the interface client's capability is encountered, the wireless link may need to send its information back to the infrastructure server, and the infrastructure server may then convert the stored scenario to fit the new interface client. This assumes that the wireless link serves the interface client (executes the



interface driver program). In the case that the interface client executes the interface driver program, the wireless link must inform the infrastructure server that a new interface client is being connected.

- 5 The following portion of the specification discusses four possible modes of operation: (1) a normal first mode, (2) a second mode where there is no external interface client, (3) a third mode where there is no infrastructure connection but there is access to a local or proximal interface client, and (4) a fourth mode where there is no connection to infrastructure and no external interface client access.

10

Figure 5 is a schematic diagram illustrating a normal mode 500 of operation in accordance with an embodiment of the present invention. In this first mode, the wireless link device 102 is connected to both the infrastructure server 104 through the wireless infrastructure 108 and to an external interface client 110 via a transceiver 116. Examples of locations where such a mode may exist include be a hotel room, in a train, in a car (with display) or at the airport. The applications are controlled by the infrastructure server 104, and the wireless link 102 is similar to a smart thin client.

15

- 20 Figure 6 is a schematic diagram illustrating a second mode 600 of operation where no external interface client is available in accordance with an embodiment of the present invention. This is very similar to the first mode in that the wireless link device is in communication with the infrastructure server 104 via the wireless network 108 except that the wireless link device 102 utilizes its own built-in interface. Examples of scenarios where this mode may exist include use of the wireless link walking on the street, sitting in a fishing boat, or driving in a car. The operation is similar to the operation in the normal mode with the application being run by the infrastructure server and the wireless link acting as a thin client.

25

- 30 Figure 7 is a schematic diagram illustrating a third mode 700 of operation where there is no infrastructure connection from the wireless link to the infrastructure

server but where there is access to an interface client proximal to the wireless link in accordance with an embodiment of the present invention. In this third mode, the wireless link **102** is not able to connect to the infrastructure but has access to an external interface client **110** via the transceiver **116**.

5

Examples of this could be in an airplane, in a car outside the wireless coverage area. In this scenario, the functionality of the wireless link **102** may be limited to built-in applications and applications downloaded from the infrastructure server **104** when the wireless link was last connected to the server. Some user data may be available

10

and can be synchronized with the main data residing at the infrastructure server upon reconnection. Some illustrative built-in applications may include scheduler/calendar and address book as well as entertainment applications like games and music

15

recording and/or playing capabilities. The external interface client **110** can be used to improve the user experience by providing a more capable visual display for the viewing the application (such as, for example, when playing a game on an airplane or when examining the user's schedule). This may not result in the absolute optimal user experience since the wireless link is controlling the interface client and may have limited memory, processing capabilities, and power supply (i.e., being battery powered).

20

Figure **8** is a schematic diagram of a fourth mode **800** of operation wherein there is no connection to infrastructure from the wireless link and no external interface client access in accordance with an embodiment of the present invention. In this scenario the wireless link **102** may act like a standard personal digital assistant (PDA) where the vital applications such as a scheduler/calendar application and an address book application as well as other entertainment applications may be available and can be accessed using the built-in interface of the wireless link **102**. This mode may be utilized in areas with no wireless coverage and no access to an external interface client.

30

It is important to understand that the system is controlled from the infrastructure server 104 and that all primary data resides on the infrastructure server (or accessible from the data store. Because of this, the data may always be up to date when the user accesses it, and it may also be possible to update data and applications automatically. It may also enable cooperation to have secure links to inside a firewall using technology like virtual private networks (VPN). This way the company solves the problem of the user having stored sensitive data on a laptop or a home computer. It also ensures that the available data and applications are always fully up to date.

One could also envision a scenario where the interface client contacts the infrastructure server using a user's wireless link. This could be to send a picture back to the infrastructure server, to report mal functions, or to inform the infrastructure server that a certain interface capability is available, and the infrastructure server can the initiate contact to the user through this device.

The wireless link can be seen as a communication engine that enables the link between the interface client and the infrastructure server whenever it is available. The communication in setting up a connection could be implemented as follows:

Establishing a connection between the infrastructure server and the wireless link may happen by the existing system protocols used in the particular wireless infrastructure. The requirement is to be able to provide a connection like TCP/IP or similar connection between the infrastructure server and the wireless link. A higher-level protocol is defined allowing the infrastructure server control the wireless link device and other interface clients connected to the wireless link using relative low bandwidth. The communication between the wireless link and an interface client is also important. Again, a standard communication protocol can be used such as Bluetooth. Bluetooth is capable of supporting the scanning for other devices, so it may be able to detect other interface client while communicating with the selected interface client.

First, the device must detect the different interface clients in its vicinity. This can be initiated either by the wireless link or by the interface client. Having the wireless link initiate the communication may drain more power from the battery powered wireless link, but having the interface clients initiate the communication is likely to result in increased interference when multiple devices are trying to contact a wireless link device. If the wireless link is the device initiating the connection, it may send out a beacon, and await the interface clients reporting their presence.

On the other hand, if the interface clients are the one initiating the connection, they may send out a beacon, and the wireless link may listen for such beacons. When a beacon is received, the wireless link may acknowledge its presence, and the two devices may start to interchange relevant information. Among other information, the interface client may report its capabilities. This can be in the form of an ID-type or directly its capabilities like display resolution, sound capability, input devices etc. It may also include information not directly related to the technical information of the interface client like its location etc. This information is reported back to the infrastructure server, or adopted by the wireless link. From this point on the infrastructure server has identified the user, the presence and capabilities of the interface client. The infrastructure server formats the interface client to fit the user's preferences using the best-suited user interface. This can be done by having the infrastructure server transfer an interface driver program to the wireless link (if it is not already there) or use an interface driver program that could be executed directly by the interface client if needed (Initially it is expected that the wireless link may execute the interface driver program, but as the system gets acceptance, the code may move to run on the interface client. )A protocol with reduced bandwidth requirements can be used to control the interface client. The infrastructure server or the wireless link may know where the user ended the last session, and can continue from there. Thus, interface roaming is achieved.

The infrastructure server can also target the user's preferences based on past user behavior (patterns) and predefined user preferences. This may enable the user

interface to appear smart, and this may enable smart applications. Since the terminal is likely to be in a fixed location, the infrastructure server may be informed about the interface client's location, and this can be included in the user interface setup as well. For example, the interface client can make suggestions to the user about sales, good restaurants etc. This may be an option that the user has the power to enable or disable.

Similar, as part of scanning for new devices, the system can also register the user in the surrounding infrastructure, and this may enable personalized greetings, dedicated services, and automatic-registration actions for example.

#### Display Roaming When an Interface Client is Already Connected to the Infrastructure Server

As previously mentioned, embodiments of the present invention may enable display roaming when an interface client already has a way to contact the infrastructure server without using (i.e., going through) the user's wireless link. In such an embodiment, the user may logon to the interface client to identify the user. Many different authentication schemes may be possible – one example could be a universal ID-card like a SIM card, and fingerprints identification (or identification by biometrics), or a User ID / password system. The wireless link may be used to perform the user identification or may include a SIM card therein for enabling the identification. However, in this embodiment, the wireless link does not have to serve as the communication link between the interface client and the infrastructure server. If the interface client does not offer a convenient input interface, the wireless link could also be used as the input device.

Figure 9 is a schematic diagram of some possible exemplary modes for interface roaming when the interface client 110 has its own connection to the server 104 separate from a connection provided through the wireless link 102 in accordance with an embodiment of the present invention. The illustrative modes illustrated in Figure 9 include a mode 902 where the connection between the interface client 110

and the server is a wired connection 904 where identification of the user is provided via the wireless link 102, a mode 906 where the connection between the interface client 110 and the server is a wired connection 904 and where identification of the user is not provided via the wireless link, and a mode 908 where the connection  
5 between the interface client 110 and the server is a direct wireless connection 910 to the infrastructure or a connection through a scatter net.

As discussed above, logon to an interface client can be performed in many ways. However, there are two distinct options. Either the user physically performs a logon  
10 procedure (initiated by the user), or the interface client detects the user e.g. by contacting the wireless link, and the user's ID back. While a central user database may exist, it may be more advantageous that the some kind of electronic identification (wireless link, Electronic ID-card etc.) be used since it can identify the user and the user's infrastructure server.

15 After the user login (verified by the infrastructure server) – the infrastructure server should already know the interface client's capabilities, so the infrastructure server can tell the interface client to generate interface the user had when last connected. The infrastructure server may reformat the user experiences to the capabilities of the  
20 new interface client.

An issue in this embodiment is the ability to end a session. Since the user is not part of the link between the infrastructure server and the interface client, the connection can exist even after a user has left the interface client. Therefore, the system must  
25 include a way to identify if the user is still using the interface client. A logoff button may be provided exist as part of the interface. An embodiment may rely on the electronic identification of the user and where the system may automatically logoff if the electronic identification is no longer presented to the interface (e.g., if the wireless link is no longer in range, or the electronic ID-card is removed, the session  
30 may automatically terminate).

### Hardware Implementation

Figure 10 illustrates an exemplary system 1000 with a plurality of components 1002 in accordance with one embodiment of the present invention. As shown, such components include a network 1004 which take any form including, but not limited to a local area network, a wide area network such as the Internet, and a wireless network 1005. Coupled to the network 1004 is a plurality of computers which may take the form of desktop computers 1006, lap-top computers 1008, hand-held computers 1010 (including wireless devices 1012 such as wireless PDA's or mobile phones), or any other type of computing hardware/software. As an option, the various computers may be connected to the network 1004 by way of a server 1014 which may be equipped with a firewall for security purposes. It should be noted that any other type of hardware or software may be included in the system and be considered a component thereof.

A representative hardware environment associated with the various components of Figure 10 is depicted in Figure 11. In the present description, the various sub-components of each of the components may also be considered components of the system. For example, particular software modules executed on any component of the system may also be considered components of the system. Figure 11 illustrates a typical hardware configuration of a workstation in accordance with one embodiment having a central processing unit 1110, such as a microprocessor, and a number of other units interconnected via a system bus 1112.

The workstation shown in Figure 11 includes a Random Access Memory (RAM) 1114, Read Only Memory (ROM) 1116, an I/O adapter 1118 for connecting peripheral devices such as disk storage units 1120 to the bus 1112, a user interface adapter 1122 for connecting a keyboard 1124, a mouse 1126, a speaker 1128, a microphone 1132, and/or other user interface devices such as a touch screen (not shown) to the bus 1112, communication adapter 1134 for connecting the workstation

to a communication network 1135 (e.g., a data processing network) and a display adapter 1136 for connecting the bus 1112 to a display device 1138.

5 An embodiment of the present invention may be written using JAVA, C, and the C++ language and utilize object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications. As OOP moves toward the mainstream of software design and development, various software solutions require adaptation to make use of the benefits of OOP. A need exists for these principles of OOP to be applied to a  
10 messaging interface of an electronic messaging system such that a set of OOP classes and objects for the messaging interface can be provided.

OOP is a process of developing computer software using objects, including the steps of analyzing the problem, designing the system, and constructing the program. An  
15 object is a software package that contains both data and a collection of related structures and procedures. Since it contains both data and a collection of structures and procedures, it can be visualized as a self-sufficient component that does not require other additional structures, procedures or data to perform its specific task. OOP, therefore, views a computer program as a collection of largely autonomous  
20 components, called objects, each of which is responsible for a specific task. This concept of packaging data, structures, and procedures together in one component or module is called encapsulation.

In general, OOP components are reusable software modules which present an  
25 interface that conforms to an object model and which are accessed at run-time through a component integration architecture. A component integration architecture is a set of architecture mechanisms which allow software modules in different process spaces to utilize each others capabilities or functions. This is generally done by assuming a common component object model on which to build the architecture.  
30 It is worthwhile to differentiate between an object and a class of objects at this point. An object is a single instance of the class of objects, which is often just called a



class. A class of objects can be viewed as a blueprint, from which many objects can be formed.

5 OOP allows the programmer to create an object that is a part of another object. For example, the object representing a piston engine is said to have a composition-relationship with the object representing a piston. In reality, a piston engine comprises a piston, valves and many other components; the fact that a piston is an element of a piston engine can be logically and semantically represented in OOP by two objects.

10

OOP also allows creation of an object that “depends from” another object. If there are two objects, one representing a piston engine and the other representing a piston engine wherein the piston is made of ceramic, then the relationship between the two objects is not that of composition. A ceramic piston engine does not make up a piston engine. Rather it is merely one kind of piston engine that has one more limitation than the piston engine; its piston is made of ceramic. In this case, the object representing the ceramic piston engine is called a derived object, and it inherits all of the aspects of the object representing the piston engine and adds further limitation or detail to it. The object representing the ceramic piston engine  
15 “depends from” the object representing the piston engine. The relationship between these objects is called inheritance.

20

When the object or class representing the ceramic piston engine inherits all of the aspects of the objects representing the piston engine, it inherits the thermal  
25 characteristics of a standard piston defined in the piston engine class. However, the ceramic piston engine object overrides these ceramic specific thermal characteristics, which are typically different from those associated with a metal piston. It skips over the original and uses new functions related to ceramic pistons. Different kinds of piston engines have different characteristics, but may have the same underlying  
30 functions associated with it (e.g., how many pistons in the engine, ignition sequences, lubrication, etc.). To access each of these functions in any piston engine

object, a programmer would call the same functions with the same names, but each type of piston engine may have different/overriding implementations of functions behind the same name. This ability to hide different implementations of a function behind the same name is called polymorphism and it greatly simplifies

5 communication among objects.

With the concepts of composition-relationship, encapsulation, inheritance and polymorphism, an object can represent just about anything in the real world. In fact, one's logical perception of the reality is the only limit on determining the kinds of things that can become objects in object-oriented software. Some typical categories are as follows:

10

- Objects can represent physical objects, such as automobiles in a traffic-flow simulation, electrical components in a circuit-design program, countries in an economics model, or aircraft in an air-traffic-control system.
- 15 • Objects can represent elements of the computer-user environment such as windows, menus or graphics objects.
- An object can represent an inventory, such as a personnel file or a table of the latitudes and longitudes of cities.
- An object can represent user-defined data types such as time, angles, and
- 20 complex numbers, or points on the plane.

20

With this enormous capability of an object to represent just about any logically separable matters, OOP allows the software developer to design and implement a computer program that is a model of some aspects of reality, whether that reality is a physical entity, a process, a system, or a composition of matter. Since the object can represent anything, the software developer can create an object which can be used as a component in a larger software project in the future.

25

If 90% of a new OOP software program consists of proven, existing components made from preexisting reusable objects, then only the remaining 10% of the new

30

software project has to be written and tested from scratch. Since 90% already came from an inventory of extensively tested reusable objects, the potential domain from which an error could originate is 10% of the program. As a result, OOP enables software developers to build objects out of other, previously built objects.

5

This process closely resembles complex machinery being built out of assemblies and sub-assemblies. OOP technology, therefore, makes software engineering more like hardware engineering in that software is built from existing components, which are available to the developer as objects. All this adds up to an improved quality of the software as well as an increased speed of its development.

10

Programming languages are beginning to fully support the OOP principles, such as encapsulation, inheritance, polymorphism, and composition-relationship. With the advent of the C++ language, many commercial software developers have embraced

15

OOP. C++ is an OOP language that offers a fast, machine-executable code.

Furthermore, C++ is suitable for both commercial-application and systems-programming projects. For now, C++ appears to be the most popular choice among many OOP programmers, but there is a host of other OOP languages, such as Smalltalk, Common Lisp Object System (CLOS), and Eiffel. Additionally, OOP capabilities are being added to more traditional popular computer programming languages such as Pascal.

20

The benefits of object classes can be summarized, as follows:

- Objects and their corresponding classes break down complex programming problems into many smaller, simpler problems.
- Encapsulation enforces data abstraction through the organization of data into small, independent objects that can communicate with each other. Encapsulation protects the data in an object from accidental damage, but allows other objects to interact with that data by calling the object's member functions and structures.

25

30

- Subclassing and inheritance make it possible to extend and modify objects through deriving new kinds of objects from the standard classes available in the system. Thus, new capabilities are created without having to start from scratch.

- 5
- Polymorphism and multiple inheritance make it possible for different programmers to mix and match characteristics of many different classes and create specialized objects that can still work with related objects in predictable ways.

- 10
- Class hierarchies and containment hierarchies provide a flexible mechanism for modeling real-world objects and the relationships among them.

- Libraries of reusable classes are useful in many situations, but they also have some limitations. For example:

- 15
- Complexity. In a complex system, the class hierarchies for related classes can become extremely confusing, with many dozens or even hundreds of classes.

- Flow of control. A program written with the aid of class libraries is still responsible for the flow of control (i.e., it must control the interactions among all the objects created from a particular library). The programmer has to decide which functions to call at what times for which kinds of objects.

- 20
- Duplication of effort. Although class libraries allow programmers to use and reuse many small pieces of code, each programmer puts those pieces together in a different way. Two different programmers can use the same set of class libraries to write two programs that do exactly the same thing but whose internal structure (i.e., design) may be quite different, depending on hundreds

25

of small decisions each programmer makes along the way. Inevitably, similar pieces of code end up doing similar things in slightly different ways and do not work as well together as they should.

Class libraries are very flexible. As programs grow more complex, more

30

programmers are forced to reinvent basic solutions to basic problems over and over

again. A relatively new extension of the class library concept is to have a framework of class libraries. This framework is more complex and consists of significant collections of collaborating classes that capture both the small scale patterns and major mechanisms that implement the common requirements and design in a specific application domain. They were first developed to free application programmers from the chores involved in displaying menus, windows, dialog boxes, and other standard user interface elements for personal computers.

Frameworks also represent a change in the way programmers think about the interaction between the code they write and code written by others. In the early days of procedural programming, the programmer called libraries provided by the operating system to perform certain tasks, but basically the program executed down the page from start to finish, and the programmer was solely responsible for the flow of control. This was appropriate for printing out paychecks, calculating a mathematical table, or solving other problems with a program that executed in just one way.

The development of graphical user interfaces began to turn this procedural programming arrangement inside out. These interfaces allow the user, rather than program logic, to drive the program and decide when certain actions should be performed. Today, most personal computer software accomplishes this by means of an event loop which monitors the mouse, keyboard, and other sources of external events and calls the appropriate parts of the programmer's code according to actions that the user performs. The programmer no longer determines the order in which events occur. Instead, a program is divided into separate pieces that are called at unpredictable times and in an unpredictable order. By relinquishing control in this way to users, the developer creates a program that is much easier to use.

Nevertheless, individual pieces of the program written by the developer still call libraries provided by the operating system to accomplish certain tasks, and the programmer must still determine the flow of control within each piece after it's called by the event loop. Application code still "sits on top of" the system.

Even event loop programs require programmers to write a lot of code that should not need to be written separately for every application. The concept of an application framework carries the event loop concept further. Instead of dealing with all the nuts and bolts of constructing basic menus, windows, and dialog boxes and then making these things all work together, programmers using application frameworks start with working application code and basic user interface elements in place. Subsequently, they build from there by replacing some of the generic capabilities of the framework with the specific capabilities of the intended application.

Application frameworks reduce the total amount of code that a programmer has to write from scratch. However, because the framework is really a generic application that displays windows, supports copy and paste, and so on, the programmer can also relinquish control to a greater degree than event loop programs permit. The framework code takes care of almost all event handling and flow of control, and the programmer's code is called only when the framework needs it (e.g., to create or manipulate a proprietary data structure).

A programmer writing a framework program not only relinquishes control to the user (as is also true for event loop programs), but also relinquishes the detailed flow of control within the program to the framework. This approach allows the creation of more complex systems that work together in interesting ways, as opposed to isolated programs, having custom code, being created over and over again for similar problems.

Thus, as is explained above, a framework basically is a collection of cooperating classes that make up a reusable design solution for a given problem domain. It typically includes objects that provide default behavior (e.g., for menus and windows), and programmers use it by inheriting some of that default behavior and overriding other behavior so that the framework calls application code at the appropriate times.

There are three main differences between frameworks and class libraries:

- Behavior versus protocol. Class libraries are essentially collections of behaviors that you can call when you want those individual behaviors in your program. A framework, on the other hand, provides not only behavior but also the protocol or set of rules that govern the ways in which behaviors can be combined, including rules for what a programmer is supposed to provide versus what the framework provides.
- Call versus override. With a class library, the code the programmer instantiates objects and calls their member functions. It's possible to instantiate and call objects in the same way with a framework (i.e., to treat the framework as a class library), but to take full advantage of a framework's reusable design, a programmer typically writes code that overrides and is called by the framework. The framework manages the flow of control among its objects. Writing a program involves dividing responsibilities among the various pieces of software that are called by the framework rather than specifying how the different pieces should work together.
- Implementation versus design. With class libraries, programmers reuse only implementations, whereas with frameworks, they reuse design. A framework embodies the way a family of related programs or pieces of software work. It represents a generic design solution that can be adapted to a variety of specific problems in a given domain. For example, a single framework can embody the way a user interface works, even though two different user interfaces created with the same framework might solve quite different interface problems.

Thus, through the development of frameworks for solutions to various problems and programming tasks, significant reductions in the design and development effort for software can be achieved. A preferred embodiment of the invention utilizes HyperText Markup Language (HTML) to implement documents on the Internet together with a general-purpose secure communication protocol for a transport

medium between the client and the Newco. HTTP or other protocols could be readily substituted for HTML without undue experimentation. Information on these products is available in T. Berners-Lee, D. Connolly, "RFC 1866: Hypertext Markup Language - 2.0" (Nov. 1995); and R. Fielding, H. Frystyk, T. Berners-Lee, J. Gettys  
5 and J.C. Mogul, "Hypertext Transfer Protocol -- HTTP/1.1: HTTP Working Group Internet Draft" (May 2, 1996). HTML is a simple data format used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of domains. HTML has been in use by  
10 the World-Wide Web global information initiative since 1990. HTML is an application of ISO Standard 8879; 1986 Information Processing Text and Office Systems; Standard Generalized Markup Language (SGML).

To date, Web development tools have been limited in their ability to create dynamic  
15 Web applications which span from client to server and interoperate with existing computing resources. Until recently, HTML has been the dominant technology used in development of Web-based solutions. However, HTML has proven to be inadequate in the following areas:

- Poor performance;
- 20 • Restricted user interface capabilities;
- Can only produce static Web pages;
- Lack of interoperability with existing applications and data; and
- Inability to scale.

25 Sun Microsystems's Java language solves many of the client-side problems by:

- Improving performance on the client side;
- Enabling the creation of dynamic, real-time Web applications; and
- Providing the ability to create a wide variety of user interface components.



With Java, developers can create robust User Interface (UI) components. Custom "widgets" (e.g., real-time stock tickers, animated icons, etc.) can be created, and client-side performance is improved. Unlike HTML, Java supports the notion of client-side validation, offloading appropriate processing onto the client for improved performance. Dynamic, real-time Web pages can be created. Using the above-mentioned custom UI components, dynamic Web pages can also be created.

Sun's Java language has emerged as an industry-recognized language for "programming the Internet." Sun defines Java as: "a simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multithreaded, dynamic, buzzword-compliant, general-purpose programming language. Java supports programming for the Internet in the form of platform-independent Java applets." Java applets are small, specialized applications that comply with Sun's Java Application Programming Interface (API) allowing developers to add "interactive content" to Web documents (e.g., simple animations, page adornments, basic games, etc.). Applets execute within a Java-compatible browser (e.g., Netscape Navigator) by copying code from the server to client. From a language standpoint, Java's core feature set is based on C++. Sun's Java literature states that Java is basically, "C++ with extensions from Objective C for more dynamic method resolution."

Another technology that provides similar function to JAVA is provided by Microsoft and ActiveX Technologies, to give developers and Web designers wherewithal to build dynamic content for the Internet and personal computers. ActiveX includes tools for developing animation, 3-D virtual reality, video and other multimedia content. The tools use Internet standards, work on multiple platforms, and are being supported by over 100 companies. The group's building blocks are called ActiveX Controls, small, fast components that enable developers to embed parts of software in hypertext markup language (HTML) pages. ActiveX Controls work with a variety of programming languages including Microsoft Visual C++, Borland Delphi, Microsoft Visual Basic programming system and, in the future, Microsoft's

development tool for Java, code named "Jakarta." ActiveX Technologies also includes ActiveX Server Framework, allowing developers to create server applications. One of ordinary skill in the art readily recognizes that ActiveX could be substituted for JAVA without undue experimentation to practice the invention.

5

Transmission Control Protocol/Internet Protocol (TCP/IP) is a basic communication language or protocol of the Internet. It can also be used as a communications protocol in the private networks called intranet and in extranet. When you are set up with direct access to the Internet, your computer is provided with a copy of the TCP/IP program just as every other computer that you may send messages to or get information from also has a copy of TCP/IP.

10

TCP/IP is a two-layering program. The higher layer, Transmission Control Protocol (TCP), manages the assembling of a message or file into smaller packet that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message. The lower layer, Internet Protocol (IP), handles the address part of each packet so that it gets to the right destination. Each gateway computer on the network checks this address to see where to forward the message. Even though some packets from the same message are routed differently than others, they'll be reassembled at the destination.

15

20

TCP/IP uses a client/server model of communication in which a computer user (a client) requests and is provided a service (such as sending a Web page) by another computer (a server) in the network. TCP/IP communication is primarily point-to-point, meaning each communication is from one point (or host computer) in the network to another point or host computer. TCP/IP and the higher-level applications that use it are collectively said to be "stateless" because each client request is considered a new request unrelated to any previous one (unlike ordinary phone conversations that require a dedicated connection for the call duration). Being stateless frees network paths so that everyone can use them continuously. (Note that

25

30

the TCP layer itself is not stateless as far as any one message is concerned. Its connection remains in place until all packets in a message have been received.).

Many Internet users are familiar with the even higher layer application protocols that use TCP/IP to get to the Internet. These include the World Wide Web's Hypertext Transfer Protocol (HTTP), the File Transfer Protocol (FTP), Telnet which lets you logon to remote computers, and the Simple Mail Transfer Protocol (SMTP). These and other protocols are often packaged together with TCP/IP as a "suite."

Personal computer users usually get to the Internet through the Serial Line Internet Protocol (SLIP) or the Point-to-Point Protocol. These protocols encapsulate the IP packets so that they can be sent over a dial-up phone connection to an access provider's modem.

Protocols related to TCP/IP include the User Datagram Protocol (UDP), which is used instead of TCP for special purposes. Other protocols are used by network host computers for exchanging router information. These include the Internet Control Message Protocol (ICMP), the Interior Gateway Protocol (IGP), the Exterior Gateway Protocol (EGP), and the Border Gateway Protocol (BGP).

Internetwork Packet Exchange (IPX) is a networking protocol from Novell that interconnects networks that use Novell's NetWare clients and servers. IPX is a datagram or packet protocol. IPX works at the network layer of communication protocols and is connectionless (that is, it doesn't require that a connection be maintained during an exchange of packets as, for example, a regular voice phone call does).

Packet acknowledgment is managed by another Novell protocol, the Sequenced Packet Exchange (SPX). Other related Novell NetWare protocols are: the Routing Information Protocol (RIP), the Service Advertising Protocol (SAP), and the NetWare Link Services Protocol (NLSP).

A virtual private network (VPN) is a private data network that makes use of the public telecommunication infrastructure, maintaining privacy through the use of a tunneling protocol and security procedures. A virtual private network can be contrasted with a system of owned or leased lines that can only be used by one company. The idea of the VPN is to give the company the same capabilities at much lower cost by using the shared public infrastructure rather than a private one. Phone companies have provided secure shared resources for voice messages. A virtual private network makes it possible to have the same secure sharing of public resources for data.

Using a virtual private network involves encryption data before sending it through the public network and decrypting it at the receiving end. An additional level of security involves encrypting not only the data but also the originating and receiving network addresses. Microsoft, 3Com, and several other companies have developed the Point-to-Point Tunneling Protocol (PPP) and Microsoft has extended Windows NT to support it. VPN software is typically installed as part of a company's firewall server.

Wireless refers to a communications, monitoring, or control system in which electromagnetic radiation spectrum or acoustic waves carry a signal through atmospheric space rather than along a wire. In most wireless systems, radio frequency (RF) or infrared transmission (IR) waves are used. Some monitoring devices, such as intrusion alarms, employ acoustic waves at frequencies above the range of human hearing.

Early experimenters in electromagnetic physics dreamed of building a so-called wireless telegraph. The first wireless telegraph transmitters went on the air in the early years of the 20th century. Later, as amplitude modulation (AM) made it possible to transmit voices and music via wireless, the medium came to be called *radio*. With the advent of television, fax, data communication, and the effective use

of a larger portion of the electromagnetic spectrum, the original term has been brought to life again.

Common examples of wireless equipment in use today include the Global  
5 Positioning System, cellular telephone phones and pagers, cordless computer accessories (for example, the cordless mouse), home-entertainment-system control boxes, remote garage-door openers, two-way radios, and baby monitors. An increasing number of companies and organizations are using wireless LAN. Wireless transceivers are available for connection to portable and notebook  
10 computers, allowing Internet access in selected cities without the need to locate a telephone jack. Eventually, it will be possible to link any computer to the Internet via satellite, no matter where in the world the computer might be located.

Bluetooth is a computing and telecommunications industry specification that  
15 describes how mobile phones, computers, and personal digital assistants (PDA's) can easily interconnect with each other and with home and business phones and computers using a short-range wireless connection. Each device is equipped with a microchip transceiver that transmits and receives in a previously unused frequency band of 2.45 GHz that is available globally (with some variation of bandwidth in  
20 different countries). In addition to data, up to three voice channels are available. Each device has a unique 48-bit address from the IEEE 802 standard. Connections can be point-to-point or multipoint. The maximum range is 10 meters. Data can be presently be exchanged at a rate of 1 megabit per second (up to 2 Mbps in the second generation of the technology). A frequency hop scheme allows devices to  
25 communicate even in areas with a great deal of electromagnetic interference. Built-in encryption and verification is provided.

Encryption is the conversion of data into a form, called a ciphertext, that cannot be easily understood by unauthorized people. Decryption is the process of converting  
30 encrypted data back into its original form, so it can be understood.

The use of encryption/decryption is as old as the art of communication. In wartime, a cipher, often incorrectly called a "code," can be employed to keep the enemy from obtaining the contents of transmissions (technically, a code is a means of representing a signal without the intent of keeping it secret; examples are Morse code and ASCII.). Simple ciphers include the substitution of letters for numbers, the rotation of letters in the alphabet, and the "scrambling" of voice signals by inverting the sideband frequencies. More complex ciphers work according to sophisticated computer algorithm that rearrange the data bits in digital signals.

10 In order to easily recover the contents of an encrypted signal, the correct decryption key is required. The key is an algorithm that "undoes" the work of the encryption algorithm. Alternatively, a computer can be used in an attempt to "break" the cipher. The more complex the encryption algorithm, the more difficult it becomes to eavesdrop on the communications without access to the key.

15 Rivest-Shamir-Adleman (RSA) is an Internet encryption and authentication system that uses an algorithm developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. The RSA algorithm is a commonly used encryption and authentication algorithm and is included as part of the Web browser from Netscape and Microsoft. It's also part of Lotus Notes, Intuit's Quicken, and many other products. The encryption system is owned by RSA Security.

25 The RSA algorithm involves multiplying two large prime numbers (a prime number is a number divisible only by that number and 1) and through additional operations deriving a set of two numbers that constitutes the public key and another set that is the private key. Once the keys have been developed, the original prime numbers are no longer important and can be discarded. Both the public and the private keys are needed for encryption /decryption but only the owner of a private key ever needs to know it. Using the RSA system, the private key never needs to be sent across the Internet.

30

The private key is used to decrypt text that has been encrypted with the public key. Thus, if I send you a message, I can find out your public key (but not your private key) from a central administrator and encrypt a message to you using your public key. When you receive it, you decrypt it with your private key. In addition to  
5 encrypting messages (which ensures privacy), you can authenticate yourself to me (so I know that it is really you who sent the message) by using your private key to encrypt a digital certificate. When I receive it, I can use your public key to decrypt it.

- 10 While various embodiments have been described above, it should be understood that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of a preferred embodiment should not be limited by any of the above described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144  
2145  
2146  
2147  
2148  
2149  
2150  
2151  
2152  
2153  
2154  
2155  
2156  
2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200